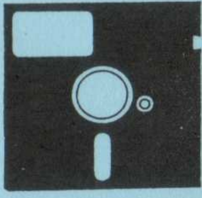
TOOLKIT IV

THE ULTIMATE DISK TOOLKIT FOR 1541/1540



instruction manual

DATEL
UTILITIES

TOOLKIT IV

© 1985

WARNING!

IF YOU DO NOT AT LEAST READ THIS PAGE YOU ARE DOOMED!

TOOLKIT IV is written for use with a single 1541 disk drive and either a C-64 or C-128 in C-64 mode. Also, TOOLKIT IV is designed to run only in countries using 50hz mains current. It will not even load if your power supply is not 50hz! While many of the utilities contained in TOOLKIT IV are fairly simple to use and self explanatory, some of the utilities are very complex and require a good deal of explanation. We have tried our best to explain what each utility does and what its purpose is in as simple terms as possible. You may find these instructions very long-winded at times, but this is only to try to make it perfectly clear to everyone. So if you do not understand, we insist you read the instructions! If you are still confused about what you are doing, you possibly don't have any reason to use that particular utility anyway. To get a clearer idea of disk operations, buy a book or two about the 1541 drive and read it. It is a good idea if you are experimenting with some of these utilities to use a scratch disk until you know exactly what you are doing. Make sure that all printers or extra drives are turned off or disconnected before loading this program. Many parts of TOOLKIT IV will not work with them turned on. If the printer is required while using TOOLKIT IV turn it on when it is needed, not before.

To load TOOLKIT IV type:

```
LOAD"*",8  : RUN
```

DATEL
UTILITIES

TABLE A: RANGE OF ZONES, TRACKS AND SECTORS

ZONE	TRACKS		LEGAL SECTORS		ILLEGAL SECTORS		DENSITY RATE
	dec	hex	dec	hex	dec	hex	$1C00 (bits 5 & 6)
1	00-17	$00-$11	0-20	$00-$14	21-25	$15-$19	3
2	18-24	$12-$18	0-18	$00-$12	19-23	$13-$17	2
3	25-30	$19-$1E	0-17	$00-$11	18-22	$12-$16	1
4	31-41	$1F-$29	0-16	$00-$10	17-21	$11-$15	0

NOTE: Track 0 and tracks 36-41 are illegal tracks and cannot be accessed normally.

TABLE B: DOS ERROR CODES

FDC	IP	DESCRIPTION	FDC	IP	DESCRIPTION
$01	00	OK	$07	25	WRITE VERIFY ERROR
$02	20	HEADER BLOCK NOT FOUND	$08	26	WRITE PROTECT ON
$03	21	NO SYNC CHARACTER	$09	27	HEADER BLOCK CHECKSUM ERROR
$04	22	DATA BLOCK NOT FOUND	$0A	28	DATA BLOCK TOO LONG
$05	23	DATA BLOCK CHECKSUM ERROR	$0B	29	DISK ID MISMATCH ERROR

TABLE C: 6510 UNDOCUMENTED OP-CODES

&AAX = and A with X, store in M
&AMA = (M & #$EE) or (A & M & #$11)
&ANA,ANB = and A with M, bit 7 to carry
&AXM = A and X minus M, to X
&AXS = (A & X) to SP, (A & X & #$11) to M
&CRA-CRL = codes that crash the processor
&DCP = decrement M, compare with A
&DNA-DNG = skip next word (two bytes)
&ISB = increment M, subtract from A
&LAN = roll M left, and with A
&LAS = load A,X,SP with M
&LAX = load A,X with M
&LOR = shift M left, or with A
&MNA-MNN = skip next byte
&NPA-NPF = nop replacements
&RAD = roll M right, add with carry
&RAM,RBM = shift (A and M) right, to A
&REO = shift M right, eor with A
&SBC = same as normal sbc #
&TEA = (A & X & >M+1) to M
&TEX = (X & >M+1) to M
&TEY = (Y & #$01) to M
&XMA = (X & M) & (A or #$EE) to A

Abbreviations

& = undocumented op-code
&,and = logical and two values
A = accumulator
X = X register
SP = stack pointer
Y = Y register
M = value referenced by operand
nop = no operation
or = logical or two values
eor = logical exclusive or two values
# = immediate mode
$ = hexadecimal number
>M+1 = high byte of operand +1

TABLE D: HEXADECIMAL CONVERSION CHART

Hex	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1-	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2-	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3-	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4-	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5-	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6-	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7-	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8-	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9-	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A-	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B-	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C-	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D-	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E-	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F-	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

END

ERROR MAKE

This will create read errors on your disks. Many commercial disks use read errors as protection. If you make a backup of one of these disks you must also reproduce the read errors as they are found on the original disk. Use ERROR LOOK first to find the read errors on the original disk.

F1 = MAKE READ ERRORS. This will bring you to the next menu. Enter the read error number you wish to create. For instance, if you want to make a 27 read error then enter 27 and hit RETURN.

20 HEADER BLOCK NOT FOUND. Enter the track and sector to be errored. Insert the DESTINATION disk into the drive and hit SPACE to write, or hit F1 to abort and return to the previous menu.

21 NO SYNC CHARACTER. Enter E to create the error on the entire track or hit P to create the error on a partial track. If you have selected Entire track then enter the track to be errored. Insert the DESTINATION disk and hit SPACE to write, or hit F1 to abort and return to the previous menu. If you have selected Partial track then enter the track and the approximate range of sectors to be errored. Insert the DESTINATION disk and hit SPACE to write, or hit F1 to abort and return to the previous menu.

22 DATA BLOCK NOT FOUND. Insert the SOURCE disk first. Enter the track and sector to be errored. The sector data will be read from the SOURCE disk and you will have the option to change the sector's current data block ID value. The normal value of the data block ID is #007. Any other value will create a DATA BLOCK NOT FOUND error. Hit RETURN if you do not want to change it. Now insert the DESTINATION disk and hit SPACE to write, or hit F1 to abort and return to the previous menu. When the error is created on the DESTINATION disk the data "underneath" the original error on the SOURCE disk will also be reproduced.

23 DATA BLOCK CHECKSUM ERROR. Insert the SOURCE disk first. Enter the track and sector to be errored. The sector data will be read from the SOURCE disk and you will have the option to change the sector's current checksum value. The checksum value varies with each sector. Hit RETURN if you do not want to change it. Now insert the DESTINATION disk and hit SPACE to write, or hit F1 to abort and return to the previous menu. When the error is created on the DESTINATION disk the data "underneath" the original error on the SOURCE disk will also be reproduced.

27 HEADER BLOCK CHECKSUM ERROR. This will create the error on an entire track only. Enter the track to be errored. Insert the DESTINATION disk and hit F1 to abort and return to the previous menu, or hit SPACE to continue. You will be asked to enter the HEADER BLOCK CHECKSUM VALUE. Any value will do, or enter the value found with ERROR LOOK. You also have the option to change the two IDs. If you do, you will create a DISK ID MISMATCH error, but "underneath" you will create a HEADER BLOCK CHECKSUM ERROR. You should normally not need to change the IDs, but the option is there. Hit RETURN if you do not want to change the IDs. The error will then be created and you will return to the previous menu.

29 DISK ID MISMATCH. This will create the error on an entire track only. Enter the track to be errored. Insert the DESTINATION disk and hit F1 to abort and return to the previous menu, or hit SPACE to continue. You will be asked to enter the new ID1 and ID2 values. The current ID values for the disk will be displayed. You must change them in order to create the error. Any other value will do, or use the values found with ERROR LOOK. The error will then be created and you will return to the previous menu.

XX = QUIT. Enter X or XX to exit this menu.

Back to the main menu . . .

F3 = WRITE SYNC MARKS. This will write bytes on a track. By writing the default values displayed on to a track you will create a mark which will confuse any track and sector copier, such as the one found in TOOLKIT IV. By placing these marks on lower numbered tracks you can effectively stop the copier in its tracks so it will not be able to continue past that track. One way to get around this, though, is to simply open the drive door long enough for the copier to continue on to the next track. Enter the track value in decimal. Then enter the actual byte value to be written in hex. Any value from $00 to $FF is acceptable. Then enter the number of times the byte is to be written on that track. Any number from 1 to 65535 is acceptable. If everything is correct hit SPACE to write, or hit F1 to return to the main menu. Note: writing these marks to the track will corrupt the track.

F5 = ERROR LOOK. This will bring you to ERROR LOOK (see next section).

F7 = DIRECTORY. This will display the directory of the disk currently in the drive.

F8 = QUIT. Hit F1 to abort and return to the main menu, or insert the TOOLKIT IV disk and hit SPACE to reload the TOOLKIT IV MAIN MENU.

TOOLKIT IV MAIN MENU

This is what you will see first when TOOLKIT IV has loaded. Keep the TOOLKIT IV disk in the drive and select any letter from A to P. Hit SPACE to load that program from the disk, or hit F1 to abort and return to the MAIN MENU. Many of the TOOLKIT IV programs allow you to reload the TOOLKIT IV MAIN MENU when finished. Others will reset the computer and you will have to reload the MAIN MENU yourself.

DOCTOR V2

DOCTOR V2 will allow you to edit any legal or illegal sector on a disk. The range of tracks and sectors is listed in TABLE A. Also sectors on renumbered tracks can be edited and rewritten. Sectors on tracks 36 and 41 can be read but usually cannot be rewritten successfully due to timing problems.

Commands:

CRSR KEYS = MOVE CURSOR. By using the CRSR keys you can move the cursor up, down, left and right within the sector edit window.

+ = NEXT SECTOR. This will read the next sector on the current track. For instance, if you have just read sector 5 then this will read sector 6.

– = PREVIOUS SECTOR. This will read the previous sector on the current track. For instance, if you have just read sector 13 then this will read sector 12.

SHIFT + = NEXT TRACK. This will read the sector on the next track. For instance, if you have just read track 7, sector 2, then this will read track 8, sector 2.

SHIFT – = PREVIOUS TRACK. This will read the sector on the previous track. For instance, if you have just read track 36, sector 0, then this will read track 35, sector 0.

R = REWRITE SECTOR. You must rewrite a sector which you have changed in the edit window in order to make the change permanent. Rewriting will also fix any READ ERROR 22 or 23 on that sector. The sector is always read again to verify after a rewrite.

T = TEXT MODE. Changes the data in the edit window starting at the current cursor position. Use any key on the keyboard and, in conjunction with CTRL, SHIFT or CBM keys, to create different values. Hit RETURN to exit edit mode.

M = MOVE SECTOR. Similar to REWRITE SECTOR, but this will rewrite the sector to a new track and sector which you input.

I = INITIALIZE DRIVE/RE-READ SECTOR. This is a very important command when using DOCTOR V2. This is useful if the drive head gets stuck for any reason. For instance, when you read track 0, the head can have problems aligning itself to another track. Use this command to fix this. This command should also be used if the Physical Track value shown is different to the Imbedded Track value. If after using this command the Physical Track value still differs from the Imbedded Track value then you can probably assume you are reading a renumbered track. Also, you must use this command if you change disks without restarting DOCTOR V2. This will reset the disk ID and allow you to work with the new disk properly.

V = VIEW AGAIN. Use this to reread the same sector. Useful if you have made a mistake changing the sector data in the edit window.

L = LINK. use this to trace through files on disk. The next sector in the file will be read. The link is indicated by bytes 0 and 1 in each sector, indicating the next track and sector respectively.

J = JUMP TO LINK UNDER CURSOR. This will read a track and sector indicated by the current cursor position. The value under the cursor will be the track value and the next byte will be the sector value.

B = BLOCK. The screen will clear and you can then enter a new track and sector to read.

Q = QUIT. This will stop the program. Hit F1 to abort and restart DOCTOR V2 or insert the TOOLKIT IV disk and hit SPACE to reload the TOOLKIT IV MAIN MENU.

H = HELP. This will give you a brief description of the DOCTOR V2 commands. Hit RETURN to exit the HELP screen.

$ = CATALOG. This will show you the latest directory of the disk currently in the drive.

@ = NEW VALUE. This will allow you to enter a new screen (not ASCII) value of the byte currently under the cursor in decimal.

A = ANOTHER DISK. This will restart DOCTOR V2.

After entering the track and sector, the 256 data bytes of that sector will be displayed on the screen in the edit window at the top half of the screen. If any error occurs, the error will be displayed at the very bottom of the screen after the letters FDC. On the bottom half of the screen will be displayed the cursor position and the value of the character under the cursor in the edit window, and below that the status of the sector:

IMB TRK tells you the actual value of the track currently under the read/write head.

PHY TRK tells you the physical position of the read/write head.

TRACK ID1 & ID2 tells you the ID of the track currently under the read/write head.

DISK ID1 & ID2 tells you the disk ID which is read from track 18, sector 0.

HDR CHK tells you the checksum of the header of the sector being read.

DB CHK tells you the data-block checksum of the sector being read. This checksum is recomputed each time the data-block is rewritten to the disk.

Link will tell you the next track and sector of the file currently being examined. If the end of the file is reached, (No Link) will be displayed.

FDC stands for Floppy Disk Controller. The FDC status is checked each time a new sector is read. If the read is successful then OK will be displayed. Otherwise an error message corresponding to the FDC status will be displayed. TABLE A lists the error message numbers and types.

DISK LOG

This will list all the files on a disk including file type, loading address, ending address, beginning track and sector, and file size. If a file is other than a PRG file then the number of bytes in the file will be displayed instead of the load and end address. By typing Y to FILE CHECK? all information will be listed. By typing N, the ending address or number of bytes will not be computed and displayed. When finished, hit F1 to restart DISK LOG or insert the TOOLKIT IV disk and hit SPACE to reload the TOOLKIT IV MAIN MENU.

BASIC COMPACTOR/UNCOMPACTOR

BASIC COMPACTOR will read a BASIC program file and decrease its size by removing all unnecessary REMs and spaces and combing lines when possible.

F1 = BEGIN COMPACTOR. Insert the disk with the file you want compacted. Enter the file name and hit RETURN. When the file is found the file will be traced through once and all jumps in the program will be recorded. Then the file will be read again and the current line number being compacted will be displayed. When finished, a compacted file with the same name as the original file but with a /C added at the end of the file name will be produced on the disk. If any errors occur during compacting the program will abort and the error message will be displayed on the screen.

F3 = BEGIN UNCOMPACTOR. This will read a BASIC program file and create a new file with each individual statement on a line of its own. Insert the disk with the file you want to uncompact. Enter the file name and hit RETURN. When the file is found the file will be read and the current line number being uncompacted will be displayed. When finished, an uncompacted file with the same name as the original file name but with a /U added at the end of the file name will be produced on the disk.

F5 = MAX LINE LENGTH. Before compacting a BASIC program hit F5 to toggle the maximum line length between 70 bytes or 250 bytes. A maximum line length of 250 bytes will create the most compact file as possible. A maximum line length of 70 bytes is best used to restore a file which has been uncompacted, or which has very few statements per line.

F7 = CATALOG. This will display the directory of the disk currently in the drive.

F8 = QUIT.

HEADER/GAP EDITOR

HEADER/GAP EDITOR will allow you to rewrite an individual sector header as well as the gap between the header and the data block. The header cannot normally be read or rewritten mainly because it is GCR encoded. This program overcomes this problem, as well as a few other problems. Any limitations will be explained.

F1 = EDIT SECTOR HEADER. Insert the disk you want to read and hit RETURN. Enter the track and physical sector to be read. The previous sector MUST be intact. For instance, if you want to read sector 5 then sector 4 must be intact in order for section 5 to be read. This is because this program syncs up to the previous sector and then counts up to the header of the sector to be read. Because of this, if the sector has been renumbered you must enter the physical location of the sector. For instance, if sector 8 has been renumbered as sector 22 you must still enter 8 as the physical sector value, not 22. When the header has been found it will be decoded and displayed on the screen.

EDIT HEADER? If you type N then you will return to the main menu. If you type Y then you can edit all the values below. The values are listed on the screen in the order which they are found on the disk.

HBID: This is always a $08. This byte indicates that the following information is a header and not a data block. Changing this to any other value will cause a HEADER NOT FOUND error for that sector.

CHKSUM: The checksum varies with each sector header. It is computed by EORing the sector value, track value and the two ID values. Normally when editing the header you should hit RETURN on this value. By hitting RETURN the value will automatically be recomputed before rewriting the header to the disk. If you change this value before hitting RETURN then the checksum will not be recomputed and instead the value you have entered will be used.

SECTOR: This is the value of the sector. By changing this value you will renumber the sector. You can change it to another legal value, or you can change it to an illegal value as listed in TABLE A, or you can change it back to its proper value if it has already been changed.

TRACK: This is the track value. By changing this you will have renumbered the track value for this particular sector only. This will create a HEADER NOT FOUND error.

ID2: Changing this value will effectively give an ID MISMATCH error for this particular sector only. But, in reality, on rereading this sector you will get a HEADER NOT FOUND error.

ID1: See ID2.

OFF1: The off-bytes in the header are not used for information but are used as padding during GCR conversion. Normally the value is $0F. Any value is all right, although we have found that the value of OFF1 should be $0F or less otherwise a HEADER NOT FOUND error can occur during rereading. The value of OFF2 can be any value.

OFF2: See OFF1.

GAP: The gap is usually either 8 or 9 bytes long depending on the ROM version of the disk drive on which the disk has been formatted. The header gap is not GCR converted, The value of the bytes found here is usually $55. When reading the gap many problems can arise because it is not GCR converted. The end of the gap is found by reading the gap until a $FF byte is found. This indicates the sync mark of the sector data block. Usually this program will successfully read this byte and only the header gap will be displayed. But some factors can cause the sync mark to be misread as a different value. If this is the case then up to another 30 bytes could be displayed here. These will be some of the following data block bytes. This is not a fault of this program but a bug in the 1541 DOS. In non-GCR mode the sync mark could be five bytes long but when read it appears as only one single byte in length. If you try to rewrite more than the number of bytes the header gap actually contains then you will destroy the sync mark and scramble the data block and a DATA BLOCK NOT FOUND error will occur when rereading the sector.

Overcoming these problems is not too hard. First, guess how long you think the header gap should be. One thing which could help would be to look at other header gaps on the disk and see how long they are. The header gap is usually the same length over the entire disk. This program will also print a red arrow if the low nibble of a byte is equal to a $F. Quite often only the high nibble of the sync mark is misread. The red arrow could indicate a possible sync mark, but not always. Another thing to look for is a single byte $55. This is the GCR data block ID. This means the previous byte could be the sync mark, or $FF. For instance, if byte 11 is equal to $55 then byte 10 could be the sync mark. Be careful not to confuse other $55 bytes for the data block ID. When you have determined the length of the header gap, place a $FF at the next value. For instance, if the header gap should be 9 bytes long, type $FF at byte number 10 and hit RETURN. The following bytes will not be edited or written to the disk. Hit F1 to abort and return to the main menu, or hit SPACE to rewrite the header gap to the disk.

When you rewrite to the disk, the track will be verified by making a simple check on each sector of the track. The error status for each sector will be displayed. At any point you can hit F1 to abort and return to the main menu. Note: as the header gap is not GCR encoded, if you change the values in the gap they may not be reread as the same values, even though they have been correctly written. This is because of the same reason why a sync mark can be misread. Writing bytes that will cause three or more zero bits to be written together can cause this problem. Also note that it is possible to read the header of sector 0 even though there is no previous sector. This program simply uses the last sector in the track for reference. But you will most likely find problems if you try to rewrite the header of sector 0. This is because the gap between the last sector and sector 0 varies each time the disk is formatted. This makes it very difficult for this program to write the new sector 0 header back to exactly the same place on the disk.

Back to the main menu . . .

F3 = EDIT SECTOR TAIL-GAP. The sector tail-gap is the gap between the end of one sector and the beginning of the next. During a normal format the drive decides what length this gap should be. By using the FORMAT program with TOOLKIT IV you can define how long the gap should be. The tail-gap can be any length, but usually is not less than 4 bytes long and not more than 8 or 9 bytes long. The tail-gap length can also change each time the sector is rewritten. Reading the tail-gap works the same way as reading the header gap except that the previous sector does not need to be intact. Instead, the sector to be read must be intact. Because the tail-gap is not GCR encoded you will find that reading the actual values of the bytes in the gap may not be successful. This is due to the problems explained earlier. This program will print up to 60 bytes of the tail-gap until it finds the sync mark of the next sector header. Sometimes, though, the sync mark will be misread, just as when reading the header gap. When this happens, use the same techniques as explained earlier to determine the actual length of the tail-gap. One difference, though, is instead of looking for a single $55 byte, you should look for a single $52 byte. This is the GCR header block ID. This indicates the first byte of the next header block, which means the previous byte should be the sync mark, or $FF. If you rewrite more than the number of bytes the tail-gap actually contains you will create a HEADER NOT FOUND error on the next sector. Hit F1 to abort and return to the main menu, or hit SPACE to rewrite the tail-gap to the disk.

Back to the main menu . . .

F5 = VERIFY A TRACK. Enter the track to be checked and hit RETURN. Each sector will be checked and the error status will be displayed. At any time you can hit F1 to abort and return to the main menu.

F7 = CATALOG. This will display the catalog of the disk currently in the drive.

F8 = QUIT. Hit F1 to abort and return to the main menu, or insert the TOOLKIT IV disk and hit SPACE to reload the TOOLKIT IV MAIN MENU.

END

FILE COMPACTOR

This will compact the file size of any machine code program. The program can sit anywhere in memory. The maximum file size which can be compacted is 202 blocks. Compacting will not affect the way the program runs and the new program will run exactly the same way as the original. Compacting a program is very useful for reducing the amount of space it takes up on a disk and can also reduce loading time. **FILE COMPACTOR will not compact basic programs.**

F1 = RESTART. Hitting F1 at any time will restart FILE COMPACTOR.

F7 = DIRECTORY. This will display the directory of the disk currently in the drive.

Place your disk with the program to be compacted on in the disk drive and then start by entering the file name of the program to be compacted, followed by the file name you wish to use for the new compacted program. Enter the value of the byte to be put in location $01 by the compacted program. Location $01 controls the I/O of the 6510. The value found in this location on power-up is $37. Refer to the Programmer's Reference Guide for more information. For most programs it will be acceptable to use $37. Next you must enter the start (SYS) address of the program being compacted. Remember that you must enter this value in hex. The program will then be read from your disk and if it compacts successfully you will be told to press SPACE to save. But first you must insert a formatted disk with at least 200 blocks free. Pressing SPACE will then save the new compacted program and automatically run the program. When you reset the computer and list the directory of the disk you will be able to see the reduced size of the new program. To run it simply use the normal syntax LOAD"FILENAME",8 and RUN.

FORMAT

FORMAT will let you easily format an entire disk in 10 seconds and also allow you to change format parameters and format individual tracks.

F1 = FORMAT ENTIRE DISK (1-35). By hitting this you will get a new menu:

F1 = NORMAL FORMAT. This will allow you to easily format a disk. Enter the disk name and enter ID1 and ID2. Note the ID values must be entered in hex. Insert the disk to be formatted. The program will not continue until it detects you have pulled a disk out of the drive and inserted another. By doing this you will be able to verify you have the correct disk to be formatted in the drive. ***Better safe than sorry.*** When you have inserted a disk, hit SPACE to format the disk or hit F1 to abort and return to the main menu. When formatting, the read/write head will bump rapidly. This is to ensure that the head is in correct alignment. This is very important. When the formatting is complete the error status and new directory will be displayed on the screen. Hit any key to return to the main menu.

F3 = SET FORMAT PARAMETERS. This next section will not go into too much detail about what each parameter does. We assume that if you are changing the parameters you will know what you are doing. The parameter values will be preceded by either a $ or a # depending on whether the value is to be entered in hex or in decimal. If neither a $ nor a # is shown then the value is to be entered in decimal. TABLE D is a hex to decimal conversion chart. Most of the parameters will allow you to enter a value from $00 to $FF. Odd things can happen if you set parameters too high or too low. For instance, setting the top sector of a track to 207 does not mean you will have sectors from 0 to 207 on that track. It is not possible to squeeze that many into a track. But the program will accept that number and will try to write all the sectors from 0 to 207 on the track. So use your own judgement.

FIRST SCREEN: The default values for each zone, track and sector are shown. Hit RETURN on each value unless you want to change it. The four zones decide the rate at which data will be written to the disk, also known as the bit-density rate or the clock rate. There are four different rates at which data can be written or read. A track formatted at a rate different than its normal rate cannot be read normally. See TABLE A for a list of tracks, sectors and density rate per zone.

SECOND SCREEN:

DISK NAME: The disk name can be up to 16 characters long.

ID1 VALUE: The two IDs will be formatted on every track on the disk. Any value from $00 to $FF can be entered.

ID2 VALUE: See ID1.

HEADER BLOCK SYNC VALUE: The sync value should always be $FF.

WRITE HEADER BLOCK SYNC: A sync mark must consist of no less than 10 consecutive 1 bits. As each byte consists of 8 bits you must write the sync value at least $02 times. A normal 1541 format writes this byte $05 times.

HEADER BLOCK ID VALUE: This byte identifies the data following the sync mark as a header. This is normally a $08.

HEADER BLOCK OFF-BYTE1 VALUE: The off-bytes in the header are unused. It is possible that some disk protection schemes use abnormal off-bytes. Values $0F or less are recommended, otherwise a HEADER NOT FOUND error may occur.

HEADER BLOCK OFF-BYTE2 VALUE: See OFF-BYTE1.

HEADER GAP BYTE VALUE: The header gap value is normally $55. Any value other than $FF is acceptable.

WRITE HEADER GAP BYTE: The header gap is usually $08 or $09 bytes long on a 1541 formatted disk depending on your ROM version.

DATA BLOCK SYNC VALUE: The sync value should always be $FF.

WRITE DATA BLOCK SYNC: Same as WRITE HEADER BLOCK SYNC.

DATA BLOCK ID VALUE: This byte identifies the data following the sync mark as a data block. This is normally a $07.

DATA BLOCK FILL-BYTE VALUE: A normal 1541 format formats each data block filled with $01s. Any value is acceptable.

BYTE 0 OF DATA BLOCK VALUE: As well as filling the data block with $01s, the normal 1541 format also adds a $4B in byte 0 of each data block. Any value is acceptable.

DATA BLOCK SIZE: The normal 1541 format sector size is #260 8-bit bytes. The range is from #001 to #260. By decreasing this value the end of each sector will be chopped off accordingly. Any value other than #260 will cause read/write problems with each sector later on.

SECTOR TAIL-GAP BYTE VALUE: The sector tail-gap value is usually $55. Any value other than $FF is acceptable.

WRITE SECTOR TAIL-GAP BYTE: On a normal 1541 formatted disk this value is variable. The value $08 is arbitrary.

ATTEMPT TO WRITE BAM IMAGE: If you type Y (yes) then a new BAM image and directory will be created. By typing N (no) it won't.

If everything is correct, insert the disk to be formatted and hit SPACE to format the disk or hit F1 to abort and return to the main menu. After formatting, the error status and new directory will be displayed on the screen. Hit any key to return to the main menu.

Back to the main menu . . .

F3 = FORMAT SINGLE TRACK. Enter the physical track to be formatted. Any track from 0 to 41 can be formatted. Also any half-track can be formatted by adding a ".5" to the track value. A half-track is a track in between two tracks. For example, track 13.5 would be found between tracks 13 and 14. Remember that if you format track 13.5, tracks 13 and 14 will be corrupted due to the wide read/write head on the 1541. Insert the disk to be formatted and hit SPACE to align the read/write head. Hit F1 to abort and return to the main menu. When the head is aligned you can then enter most of the parameters as listed above. Also some new parameters have been added:

TOP SECTOR VALUE: The normal value for the top sector in the current zone will be displayed in decimal. Remember this is not the number of sectors, but the highest sector value. Sector 0 is always the lowest sector.

BIT-DENSITY RATE: The normal value for the bit-density rate in the current zone will be displayed. The value shown is the value of bits 5 and 6 of byte $1C00 in the disk drive. By changing this value the track cannot be read normally.

ID1 VALUE: The value shown here will be the value of the disk ID. By changing this value a DISK ID MISMATCH will appear on this track.

ID2 VALUE: See ID1 VALUE.

TRACK VALUE: The physical value of the track will be displayed in decimal. By changing this value you will create a renumbered track. Some protection schemes look for this. You can also change the value of the track to an illegal value. By doing this you could actually create, for instance, a track 100. The range is #000 to #255.

If everything is all right the drive motor will start up. The motor must be up to speed before formatting the track. Hit SPACE to format the track or hit F1 to abort and return to the main menu. When the formatting is complete the error status and directory will be displayed on the screen. Hit any key to return to the main menu.

Back to the main menu . . .

F3 = DISK ERASE. This will erase all data on the disk. Insert the disk to be erased. Hit SPACE to erase the disk or hit F1 to abort and return to the main menu. The current track being erased will be displayed. Hitting F1 at any time will abort and the rest of the disk will not be erased.

F6 = DRIVE SPEED TEST. This will determine the speed of the rotation of your disk drive. Ten samples will be made and then averaged together. The drive speed should average around 300 rpm.

F7 = DIRECTORY. This will display the directory of the disk currently in the drive.

F8 = EXIT FORMAT MENU. Insert the TOOLKIT IV disk and hit SPACE to return to the TOOLKIT IV MAIN MENU, or hit F1 to abort and return to the FORMAT main menu.

FORMATTING EXTRA SECTORS: By changing the following parameters you will be able to format tracks 18 to 24 with an extra sector. There is just enough room in this zone to squeeze in a sector 19. Remember you cannot access this sector normally or by using U1 or U2 commands. Enter all the other parameters as normal.

FIRST SCREEN:

	TOP TRACK	TOP SECTOR
ZONE 2	18-24	000-019

SECOND SCREEN:

WRITE HEADER BLOCK SYNC **$04**
WRITE HEADER GAP BYTE.................................**$08**
WRITE DATA BLOCK SYNC **$04**
WRITE SECTOR TAIL-GAP BYTE **$04**

END

DISK MON

DISK MON allows you to read and write to the memory in the disk drive. It operates much the same as a regular machine language monitor.

@ = DISK ERROR CHANNEL. You can read the error status of the drive or send commands to the drive. For instance:

@ will display the current drive error status.
@I0: will initialize the drive.

D = DISASSEMBLE.

D AAAA will disassemble the disk drive memory starting at address AAAA.

A = ASSEMBLE.

This will allow you to assemble routines in the drive's RAM using the standard assembler syntax.

G = GO.

G AAAA will execute a routine in the disk drive memory at address AAAA.

M = MEMORY.

M AAAA will display the disk drive memory in hex and ASCII starting at address AAAA.

T = TRANSFER.

TD AAAA BBBB will transfer 256 bytes from the computer's memory starting at address AAAA to the disk drive's memory starting at address BBBB.

TC AAAA BBBB will transfer 256 bytes from the disk drive's memory starting at address AAAA to the computer's memory starting at address BBBB.

: = ENTER MEMORY.

: AAAA BB CC DD EE FF GG HH II enters eight hex bytes in the disk drive's memory starting at address AAAA.

X = EXIT.

FILE COPY V2

FILE COPY V2 is a file copier which copies files approximately five times faster than normal. This program may not run correctly if you own a 64 with an older ROM version. This is due to problems with the colour memory.

F1 = COPY FILES. Insert the SOURCE disk and hit F1. All the file names will be read into memory and they will then be displayed on the screen. Use the CRSR up and down to move through the file names. This allows you to select the order in which the files will be copied. To select a file to copy move the file name so it is under the grey bar in the centre of the screen, then hit RETURN. Do the same with each file you want to copy. The order in which they will be copied will be printed alongside each file name under the word ORDER. The maximum file size copyable is 223 blocks.

A = ABORT. If you decide not to copy any files from this disk hit A and the program will restart.

B = BEGIN COPY. When you have selected all the files you want to copy hit B. As many files as possible will then be loaded into memory. You may have to copy all the files in more than one pass. When the files have loaded, insert a formatted DESTINATION disk and hit SPACE. If any disk error occurs it will be printed on the screen. You will have the option to try again, skip that file, save the whole buffer again, view the directory, send a disk command, or abort and restart the program. When all files have been saved, you will have the option to save all the files again to another disk. Type Y (yes) or N (no). If more than one pass is needed to copy all the files you will be asked to reinsert the SOURCE disk and hit SPACE. The rest of the files will then be loaded into memory as before.

C = CHOOSE AGAIN. If you find you have not chosen the files to copy in the order you like, hit C to start again.

Back to the main menu . . .

F3 = DIRECTORY. This will display the directory of the disk currently in the drive. Hit SPACE to return to the main menu.

F5 = DISK COMMAND. Allows you to send a command to the disk drive. This program includes a fast disk format routine. Use the normal format syntax. Example: N0: diskname,id (quotes are not necessary). Wait until the drive completely finishes formatting before continuing.

F7 = DISK STATUS. Allows you to read the current status of the disk drive. Hit SPACE to return to the main menu.

F8 = QUIT. This will reset the computer. You can reload TOOLKIT IV if necessary.

END

DISK LOOK

Insert the disk you want to look at and hit any key. The directory will be read and printed on the screen. A yellow arrow at the top of the screen can be moved through the directory by using the CRSR keys. The directory will scroll up and down. By using the CRSR RIGHT key you can move the arrow to the file type on the right of the file name. Now by using the CRSR UP and DOWN you can change the file type of the file. Hitting the asterisk (*) will toggle between a normal file type and an unclosed file type. Hitting the less-than symbol (<) will toggle between a normal file type and a locked file type. Hit RETURN to write any changes to the disk, or hit CRSR LEFT key to return to the directory mode and the file type will change back to what it was originally.

C = **NEW DISK.** This will restart DISK LOOK.

N = **DISK NAME.** This will read the disk name and display it at the top of the screen. Change the disk name and hit RETURN to write the new name to the disk. Hit STOP to abort.

£ = **DOS VERSION.** This will read the value of the DOS version found on track 18, sector 0, byte 2. The normal value for a 1541 formatted disk is 65. By changing it to any other value will cause the disk to be "write-protected". You will no longer be able to scratch any files or save any files to that disk. By changing the DOS version back to 65 you will be able to write to the disk again.

← = **SORT DIRECTORY.** You will be asked ARE YOU SURE? Type Y (yes) or N (no). The directory will be sorted into alphabetical order first by file type and then by file name. Any deleted files will be placed at the end of the directory. When sorting is finished DISK LOOK will automatically reread the directory and show you the new directory.

$ = **DIRECTORY.** This will display the normal directory of the disk. Hold SHIFT to pause, hit STOP to stop.

V = **VALIDATE DISK.** This will fix any BAM problems. It will allocate only the blocks used by the files on the disk.

H = **HELP SCREEN.** This will show you a brief summary of the DISK LOOK commands.

Q = **QUIT.** This will stop DISK LOOK and reset the computer. Reload TOOLKIT IV if necessary.

B = **VIEW/EDIT BAM.** The BAM of the disk will be read. On the left of the screen going from bottom to top will be the numbers 0 to 20. This is a guide for the number of sectors. From left to right will be the numbers 1 to 35 indicating the tracks. The screen will fill with 683 coloured crosses. Each cross represents a specific block (also called a sector) on the disk. Read this display like an x-y graph. For instance track 5, sector 9 can be found by following the number 5 along the bottom of the screen then counting up until you are also on the same horizontal line as the number 9 on the left hand side of the screen. A dark blue cross means that sector is free, or unused. A cyan reversed cross means that block is allocated. There is a maximum of 664 blocks free on a 1541 formatted disk. All of track 18 is reserved for the BAM and the directory. Notice that there are fewer sectors on higher numbered tracks.

E = **EDIT BAM.** Use the commands listed below. CRSR keys move the cursor around the BAM edit screen.

+ = Allocate the block underneath the cursor.
- = Free the block underneath the cursor.
A = Allocate all 683 blocks.
F = Free all 683 blocks.
STOP = Exit BAM edit mode without rewriting the new BAM to the disk and return to DISK LOOK.
RETURN = Write the new BAM to the disk and return to DISK LOOK.
F1 = Return to DISK LOOK.

The following commands will work on the file the yellow arrow is pointing at.

Z = **READ TEXT FILE.** This will read the file as a text file. Useful for reading wordprocessor files on the disk. Hold SHIFT to pause, hit STOP to stop.

L = **LIST BASIC PROGRAM.** This will list the file as a BASIC program. Hold SHIFT to pause, hit STOP to stop.

D = **DISASSEMBLE FILE.** This will disassemble the file as machine code. You can also print out to a printer on device 4. Hold SHIFT to pause, hit STOP to stop.

SHIFT D = DISASSEMBLE FILE WITH UNDOCUMENTED OPCODES. This will disassemble the file as machine code and include any 6510 undocumented opcodes listed in TABLE C. You can also print out to a printer on device 4. Hold SHIFT to pause, hit STOP to stop.

A = **START & END ADDRESS.** This will display the file start and end address in hex and decimal. If the file is not a PRG type then it will display the number of bytes in the file.

R = **RENAME FILE.** Enter the new file name in place of the original file name and hit RETURN to write it to the disk. Hit STOP to abort.

K = **KILL DIRECTORY ENTRY.** This will scratch the file and also clear the file name, file size and file type.

E = **EXCHANGE TWO FILES.** Move the arrow to one of the files to be exchanged and hit E. Then move the arrow to the other file to be exchanged and hit E. The two will be exchanged. Exchanging a file with itself will abort this function.

RECOVERING A SCRATCHED FILE WITH DISK LOOK. Move the arrow to the scratched file. Change the file type to its original type and remove the asterisk from the front of the file type by hitting the asterisk. Hit RETURN and the file will be recovered and the blocks used by the file will be allocated in the BAM. If the file cannot be completely recovered the message CHAIN BROKEN ERROR will be displayed.

FAST DISK COPY

FAST DISK COPY will copy an entire, unprotected disk with one drive in just under two minutes. It can also copy only the BAM of a disk and make a copy in even less time. The disk you wish to copy is called the SOURCE disk. The disk which you want to copy the SOURCE disk on to is called the DESTINATION disk. When FAST DISK COPY is running the drive will be spinning and also at times the red LED on the front of the drive may be on. This is normal. Do not be afraid to pull disks in and out of the drive when requested to by the program. You cannot damage anything! The drive is kept running so it is always up to speed. This saves quite a few seconds during copying.

COPY BAM? The first question asked by this program is whether you want to copy the entire SOURCE disk or copy the BAM of the SOURCE disk. Type either Y (yes) to copy BAM or N (no) then hit RETURN. To copy the BAM means to copy just the blocks allocated by the Block Availability Map which is located on track 18, sector 0. On a normal file disk many of the blocks are not used. By copying only the used blocks on the disk you can save copying time.

TRY TO READ BAD SECTORS? You may have a disk which the drive finds hard to read, possibly because it was formatted on a drive slightly out of alignment with yours or perhaps the disk itself is old or dirty. By typing Y (RETURN) this program will give a few extra tries to read any bad sector. By typing N (RETURN) this program will use its normal read mode. If any read error occurs during copying it will be printed on the screen.

VERIFY COPY? If you type Y (RETURN) each sector will be verified on the DESTINATION disk after it has been written to. If any read error occurs it will be printed on the screen.

FORMAT DISK? If your DESTINATION disk has not already been formatted then it can be formatted during copying by typing Y (RETURN). Type N (RETURN) if you do not need to format the DESTINATION disk.

DISK NAME? You may change the name of the DESTINATION disk at this point. For instance, if the SOURCE disk is called FRED you can change the name of the DESTINATION disk to BOB by typing BOB (RETURN). If you want the DESTINATION disk to have the same name as the SOURCE disk then just hit RETURN.

ID? If you have chosen to format the DESTINATION disk during copying you will have this option to change the ID of the DESTINATION disk. For instance, if the SOURCE disk ID is 4C you could change the ID of the DESTINATION disk to 6H by typing 6H (RETURN). If you want the DESTINATION disk to have the same ID as the SOURCE disk then just hit RETURN.

After you have set all the above options you will then be asked to insert the SOURCE disk into the drive. When you have done this hit RETURN. The screen will go black and the LED on the disk drive will flash rapidly. When the screen reappears any read errors that have occurred will be displayed. You are then asked to insert the DESTINATION disk. Take out the SOURCE disk and insert the DESTINATION disk. Hit RETURN again and the screen will go black again. Continue to swap disks until the final message (END OF COPY ✻) is displayed. If you hit RETURN at this point the program will restart. Then if you want to copy another disk you can enter new options. If you want to copy another disk with the same options as before, hit SPACE, then hit RETURN. You will then be asked to insert the SOURCE disk and you can then proceed as you have before.

$ = CATALOG. At any point typing $ (RETURN) will display the directory of the disk currently in the drive. Notice it is not a normal directory, but the filename, size and beginning track and sector of the first 16 files will be displayed (including deleted files). If you then type Y (RETURN) the next 16 will be displayed, and so on. When you have finished reading the directory, hitting RETURN will return you to the point where you were before.

✻ = RESTART. At any point typing ✻ (RETURN) will restart the program.

There is no QUIT option on FAST DISK COPY. When you have finished copying, reset the computer and the drive. Reload TOOLKIT IV if necessary.

FILE COPY V1

FILE COPY V1 is similar to the file copier in the previous TOOLKITS. A fast load and save routine has been added to this version which will make it copy approximately three times faster than normal.

Before you select any files you can use pattern matching to display only certain files in the directory by using the asterisk (✻) and/or the question mark (?) in combination with a file name. Use an asterisk to display all files in the directory. For examples of pattern matching read the 1541 instruction manual. The maximum file size that can be copied is 210 blocks. You cannot copy REL or DEL files with a file copier. You will have to use DISK COPY for those. You select files you wish to copy from the SOURCE disk by selecting Y (yes), N (no), A for all the files from that point in the directory onwards, or D if you are done selecting files. The files will be loaded from the disk. Now insert a formatted DESTINATION disk and hit any key to save the files to the new disk. Hit F1 if you wish to verify the saving of each file. Usually this is not necessary. When finished you have the option to save the files to another disk. Type Y to resave. If you type N then the program will end. Type RUN (RETURN) to restart. You can also reload TOOLKIT IV if necessary.

ERROR LOOK

ERROR LOOK will find all disk read errors and will also find extra or renumbered sectors, extra or renumbered tracks and can also detect half-tracks or any errors on a half-track. Insert the disk to be looked at and hit any key. You will then see the ERROR LOOK menu:

F1 = LOOK AT ALL TRACKS (1-35). This will look at all legal tracks from track 1 to 35. It will not look at extra or half-tracks. As well as looking for errors, it will look for illegal sectors on each track. A NOT FOUND, OK message will indicate that no illegal sector has been found, which is normal. A SECTOR FOUND, OK message indicates an illegal sector has been found. The range of illegal sectors can be found in TABLE A. Sectors numbered higher than the values in this table will not be looked for because the DOS prevents them from being read under any circumstances. Before each error look the drive will bump the read/write head very quickly. *Don't panic.* This is done to ensure that the head is in alignment and is important for correct reading. Hitting F1 at any time will abort the error look. When the error look has finished you have the option to list all details of each error found. You may also output to a printer on device 4. The error number and type will be printed first, followed by the physical track and sector value where the error was found. Example:

21,NO SYNC CHARACTER,05,13

This shows a 21 error was found on track 5, sector 13. Below this will be displayed the DATA BLOCK ID value, DATA BLOCK CHECKSUM value, ID1 value, ID2 value, and HEADER CHECKSUM value. Also the actual value of the TRACK and SECTOR found in the header will be displayed. A different track value than the physical track value will indicate a renumbered track. If an error 21 is found, no other header information will be displayed other than the error message. Holding down the SHIFT key at any time will pause the listing. Releasing the SHIFT key will continue the listing. When the listing has finished, hit SPACE to restart ERROR LOOK or hit F1 to relist the same errors again.

F2 = SELECT TRACKS. This will allow you to select any range of tracks to look at. Enter the first track to look at, then enter the last track to look at. If in half-track mode you must add a .5 at the end of each track value. Because of head alignment problems, track 0 should be looked at on its own. The ERROR LOOK in this mode operates exactly the same.

F3 = LOOK AT HALF TRACKS. Hit F3 to toggle between half-track mode and normal track mode. If YES is selected then only half-tracks can be looked at and you must use the SELECT TRACKS option to look at tracks. If NO is selected then you are in normal track mode and you can view normal tracks only. If viewing half-tracks, it is best to only view the necessary tracks. You will get many miscellaneous error readings on other tracks otherwise. See the section on TRACK SCAN below.

F4 = ANOTHER DISK. This will restart ERROR LOOK.

F5 = ERROR MAKE. This will bring you back to the ERROR MAKE part of the program.

F6 = SCAN TRACKS. This will scan tracks on the disk. This is especially useful for finding half-tracks. A half-track is found by using your own judgement. This is explained later. You will begin with the read/write head on track 18.0. The program will constantly check and update the current track value found under the head and also display the current error message.

F1 = MOVE HEAD UP. This will move the read/write head by half-track increments up in value.

F3 = RESTART TRACK SCAN. This will stop the scan and restart with the read/write head back on track 18.0.

F4 = BUMP HEAD, RESTART. This will first bump the read/write head then restart with the head back on track 18.0. This is useful to put the head into alignment. This is very rarely necessary.

F7 = MOVE HEAD DOWN. This will move the read/write head by half-track increments down in value.

F8 = ABORT. This will stop the track scan and return you to the ERROR LOOK menu.

When looking for half-tracks you must use your own judgement to decide whether the half-track exists or not. When using TRACK SCAN a half-track will show as a constant 00,OK error message while the read/write head is on a half-track. Approach this track from above and below to confirm this. The absence of a half-track will show a number of non-constant error messages. If looking at a commercial disk you could be wasting your time. Half-tracking as a means of protection is unknown at the moment. This is because it really doesn't offer any real protection. Half-tracking is really only a gimmick, nothing more. There is no real advantage to using half-tracks.

When using TRACK SCAN the read/write head can get confused. Sometimes it will move in the wrong direction than it should. You will be able to tell this when the physical track value displayed and the TRACK AS READ will not constantly be the same values. If this happens, use F3 to restart and try again.

Back to the main menu . . .

F1 = DIRECTORY. This will display the directory of the disk currently in the drive.

F8 = QUIT. Hit F1 to abort and return to ERROR LOOK or insert the TOOLKIT IV disk and hit SPACE to reload the TOOLKIT IV MAIN MENU.

DATEL
UTILITIES

FENTON INDUSTRIAL ESTATE,
DEWSBURY ROAD, fenton,
STOKE-ON-TRENT TEL: 0782 273815
FAX: (0782) 264510